class 12

```r
library(BiocManager)
library(DESeq2)
```

```r
counts <- read.csv("airway_scaledcounts.csv", row.names=1)
metadata <-  read.csv("airway_metadata.csv")
```

```r
nrow(counts)
```

```
[1] 38694
```

```r
metadata
```

```
          id     dex celltype     geo_id
1 SRR1039508 control   N61311 GSM1275862
2 SRR1039509 treated   N61311 GSM1275863
3 SRR1039512 control  N052611 GSM1275866
4 SRR1039513 treated  N052611 GSM1275867
5 SRR1039516 control  N080611 GSM1275870
6 SRR1039517 treated  N080611 GSM1275871
7 SRR1039520 control  N061011 GSM1275874
8 SRR1039521 treated  N061011 GSM1275875
```

Q1. How many genes are in this dataset? 38694

Q2. How many 'control' cell lines do we have? 4

Q3. How would you make the above code in either approach more robust? Following the below code instead the class website code

```r
#control <- metadata[metadata[,"dex"]=="control",]
#control.counts <- counts[ ,control$id]
#control.mean <- rowSums( control.counts )/4
#head(control.mean)
```

```r
control.inds <- metadata$dex=="control"
control.ids <- metadata[control.inds,"id"]
control.counts <- counts[,control.ids]
head(control.counts)
```

```
                SRR1039508 SRR1039512 SRR1039516 SRR1039520
ENSG00000000003        723        904       1170        806
ENSG00000000005          0          0          0          0
ENSG00000000419        467        616        582        417
ENSG00000000457        347        364        318        330
ENSG00000000460         96         73        118        102
ENSG00000000938          0          1          2          0
```

```r
control.mean <- rowMeans(control.counts)
head(control.mean)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
         900.75            0.00          520.50          339.75           97.25
ENSG00000000938
           0.75
```

Q4. Follow the same procedure for the treated samples (i.e. calculate the mean per gene across drug treated samples and assign to a labeled vector called treated.mean)

```r
treated.inds <- metadata$dex=="treated"
treated.ids <- metadata[treated.inds,"id"]
treated.counts <- counts[,treated.ids]
head(treated.counts)
```

```
                SRR1039509 SRR1039513 SRR1039517 SRR1039521
ENSG00000000003        486        445       1097        604
ENSG00000000005          0          0          0          0
ENSG00000000419        523        371        781        509
```

```
ENSG00000000457        258         237         447         324
ENSG00000000460         81          66          94          74
ENSG00000000938          0           0           0           0
```

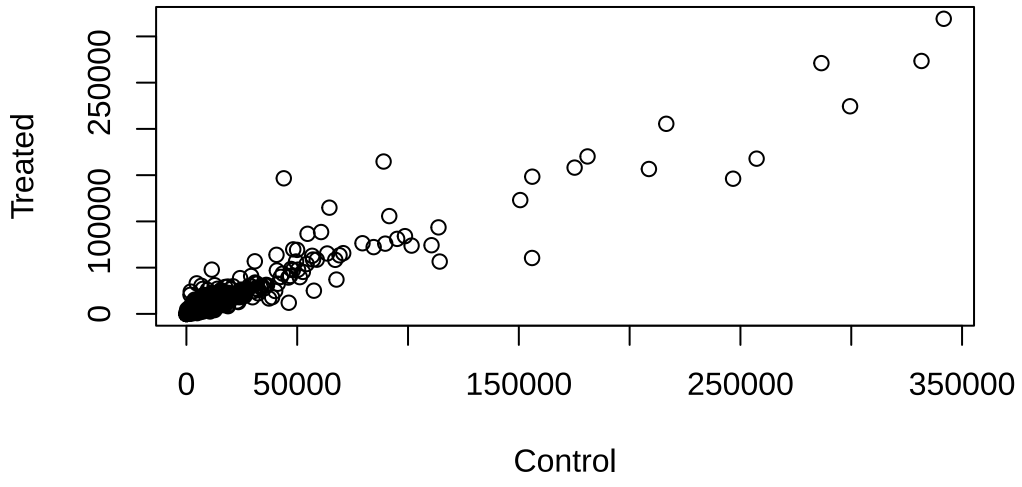
```
treated.mean <- rowMeans(treated.counts)
head(treated.mean)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
         658.00            0.00          546.00          316.50           78.75
ENSG00000000938
           0.00
```

```
meancounts <- data.frame(control.mean, treated.mean)
```

Q5 (a). Create a scatter plot showing the mean of the treated samples against the mean of the control samples. Your plot should look something like the following.

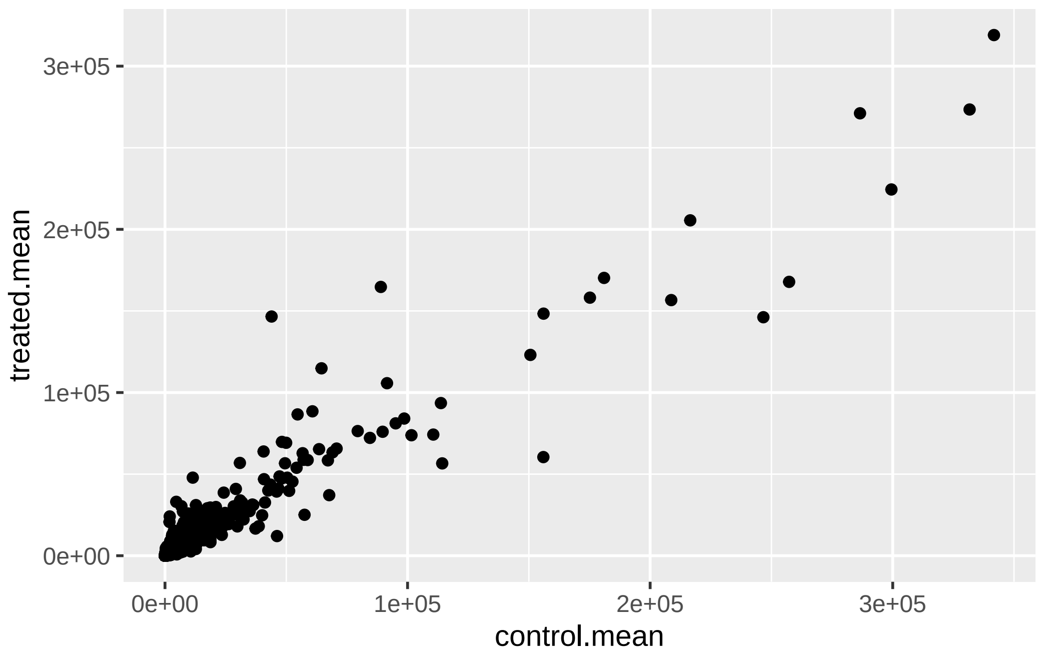
```
plot(meancounts,xlab="Control",ylab="Treated")
```



Q5 (b).You could also use the ggplot2 package to make this figure producing the plot below. What geom_?() function would you use for this plot? geom_point

```
library(ggplot2)
```

```
ggplot(meancounts)+aes(control.mean,treated.mean)+geom_point()
```

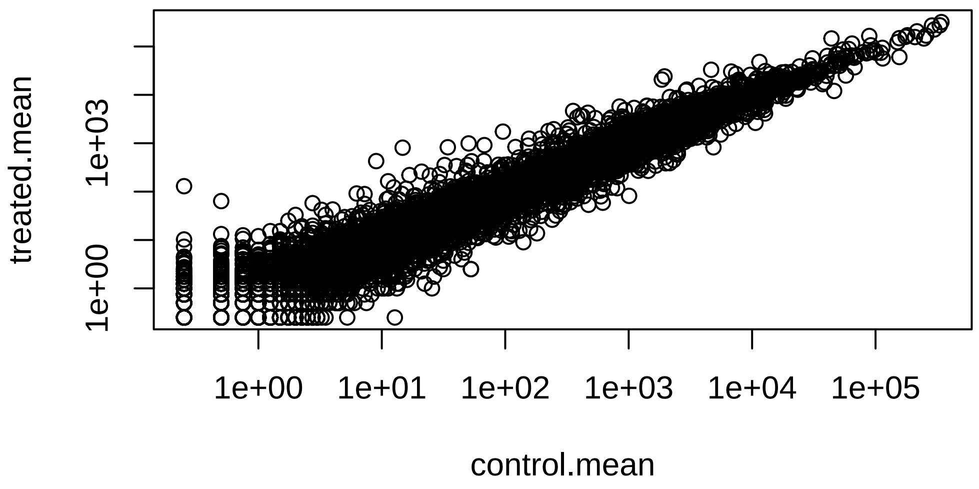


Q6. Try plotting both axes on a log scale. What is the argument to plot() that allows you to do this? log

```
plot(meancounts,log="xy")
```

```
Warning in xy.coords(x, y, xlabel, ylabel, log): 15032 x values <= 0 omitted
from logarithmic plot
```

```
Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted
from logarithmic plot
```



```r
meancounts$log2fc <- log2(meancounts[,"treated.mean"]/meancounts[,"control.mean"])
head(meancounts)
```

```
                control.mean treated.mean      log2fc
ENSG00000000003       900.75       658.00 -0.45303916
ENSG00000000005         0.00         0.00         NaN
ENSG00000000419       520.50       546.00  0.06900279
ENSG00000000457       339.75       316.50 -0.10226805
ENSG00000000460        97.25        78.75 -0.30441833
ENSG00000000938         0.75         0.00        -Inf
```

Q7. What is the purpose of the arr.ind argument in the which() function call above? Why would we then take the first column of the output and need to call the unique() function? The arr.ind will cause which() to return all the TRUE values in both the row and column. Calling unique() will ensure we dont count any row twice if it has zer entries in both samples.

```r
#zero.vals <- which(meancounts[,1:2]==0, arr.ind=TRUE)

#to.rm <- unique(zero.vals[,1])
#mycounts <- meancounts[-to.rm,]
#head(mycounts)
```

```r
to.keep.inds <- rowSums(meancounts[,1:2] == 0) ==0
mycounts <- meancounts[to.keep.inds,]
nrow(mycounts)
```

```
[1] 21817
```

Q8. Using the up.ind vector above can you determine how many up regulated genes we have at the greater than 2 fc level? 250

Q9. Using the down.ind vector above can you determine how many down regulated genes we have at the greater than 2 fc level? 367

Q10. Do you trust these results? Why or why not? fold change can be large (e.g. »two-fold up- or down-regulation) without being statistically significant (e.g. based on p-values

```r
up.ind <- mycounts$log2fc > 2
down.ind <- mycounts$log2fc < (-2)
sum(up.ind)
```

```
[1] 250
```

```r
sum(down.ind)
```

```
[1] 367
```

```r
sum(mycounts$log2fc >= 2)
```

```
[1] 314
```

```r
library(DESeq2)
```

```r
dds <- DESeqDataSetFromMatrix(countData=counts,
                              colData=metadata,
                              design=~dex)
```

```
converting counts to integer mode
```

```
Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
design formula are characters, converting to factors
```

```r
dds
```

```
class: DESeqDataSet
dim: 38694 8
metadata(1): version
assays(1): counts
rownames(38694): ENSG00000000003 ENSG00000000005 ... ENSG00000283120
  ENSG00000283123
rowData names(0):
colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521
colData names(4): id dex celltype geo_id
```

```r
dds <- DESeq(dds)
```

```
estimating size factors

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

final dispersion estimates

fitting model and testing
```

```r
res <- results(dds)
res <- as.data.frame(res)
#res
```

```r
summary(res)
```

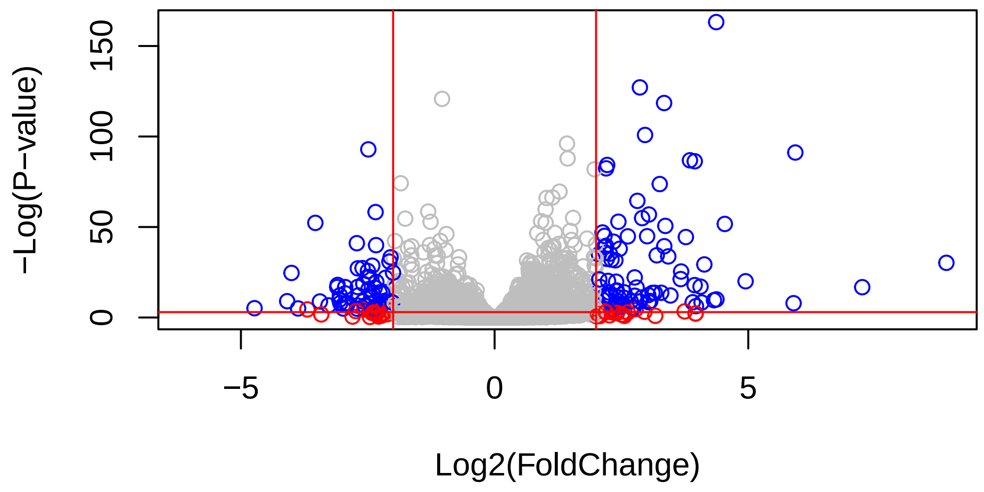
```
   baseMean         log2FoldChange       lfcSE             stat
Min.   :     0.0   Min.   :-6.030   Min.   :0.057   Min.   :-15.894
1st Qu.:     0.0   1st Qu.:-0.425   1st Qu.:0.174   1st Qu.: -0.643
Median :     1.1   Median :-0.009   Median :0.445   Median : -0.027
Mean   :   570.2   Mean   :-0.011   Mean   :1.136   Mean   :  0.045
3rd Qu.:   201.8   3rd Qu.: 0.306   3rd Qu.:1.848   3rd Qu.:  0.593
Max.   :329280.4   Max.   : 8.906   Max.   :3.534   Max.   : 18.422
                   NA's   :13436    NA's   :13436   NA's   :13436
    pvalue           padj
Min.   :0.000   Min.   :0.000
1st Qu.:0.168   1st Qu.:0.203
Median :0.533   Median :0.606
Mean   :0.495   Mean   :0.539
3rd Qu.:0.800   3rd Qu.:0.866
Max.   :1.000   Max.   :1.000
NA's   :13578   NA's   :23549
```

```r
#plot( res$log2FoldChange,  res$padj)
```

```r
mycols <- rep("grey", nrow(res))
mycols[abs(res$log2FoldChange) > 2]  <- "red"

inds <- (res$padj < 0.01) & (abs(res$log2FoldChange) > 2)
mycols[inds] <- "blue"


plot( res$log2FoldChange,  -log(res$padj),  col=mycols,
      xlab="Log2(FoldChange)",
      ylab="-Log(P-value)")
abline(v=c(-2,2), col="red")
abline(h=-log(0.05), col="red")
```



```r
library("AnnotationDbi")
library("org.Hs.eg.db")
```

```r
columns(org.Hs.eg.db)
```

```
 [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
 [6] "ENTREZID"     "ENZYME"       "EVIDENCE"     "EVIDENCEALL"  "GENENAME"
[11] "GENETYPE"     "GO"           "GOALL"        "IPI"          "MAP"
[16] "OMIM"         "ONTOLOGY"     "ONTOLOGYALL"  "PATH"         "PFAM"
[21] "PMID"         "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"
[26] "UNIPROT"
```

```r
res$symbol <- mapIds(org.Hs.eg.db,
                     keys=row.names(res),
                     keytype="ENSEMBL",
                     column="SYMBOL",
                     multiVals="first")
```

```
'select()' returned 1:many mapping between keys and columns
```

```
res$entrez <- mapIds(org.Hs.eg.db,
                     keys=row.names(res),
                     keytype="ENSEMBL",
                     column="ENTREZID",
                     multiVals="first")
```

```
'select()' returned 1:many mapping between keys and columns
```

```
res$genename <- mapIds(org.Hs.eg.db,
                     keys=row.names(res),
                     keytype="ENSEMBL",
                     column="GENENAME",
                     multiVals="first")
```

```
'select()' returned 1:many mapping between keys and columns
```

```
head(res)
```

```
                   baseMean log2FoldChange     lfcSE       stat     pvalue
ENSG00000000003 747.1941954    -0.35070302 0.1682457 -2.0844697 0.03711747
ENSG00000000005   0.0000000             NA        NA         NA         NA
ENSG00000000419 520.1341601     0.20610777 0.1010592  2.0394752 0.04140263
ENSG00000000457 322.6648439     0.02452695 0.1451451  0.1689823 0.86581056
ENSG00000000460  87.6826252    -0.14714205 0.2570073 -0.5725210 0.56696907
ENSG00000000938   0.3191666    -1.73228897 3.4936010 -0.4958463 0.62000288
                     padj   symbol entrez
ENSG00000000003 0.1630348   TSPAN6   7105
ENSG00000000005        NA     TNMD  64102
ENSG00000000419 0.1760317     DPM1   8813
ENSG00000000457 0.9616942    SCYL3  57147
ENSG00000000460 0.8158486 C1orf112  55732
ENSG00000000938        NA      FGR   2268
                                                                    genename
ENSG00000000003                                                tetraspanin 6
ENSG00000000005                                                  tenomodulin
ENSG00000000419 dolichyl-phosphate mannosyltransferase subunit 1, catalytic
ENSG00000000457                                     SCY1 like pseudokinase 3
ENSG00000000460                               chromosome 1 open reading frame 112
ENSG00000000938                    FGR proto-oncogene, Src family tyrosine kinase
```

```r
library(pathview)
```

```
##############################################################################
Pathview is an open source software package distributed under GNU General
Public License version 3 (GPLv3). Details of GPLv3 is available at
http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to
formally cite the original Pathview paper (not just mention it) in publications
or products. For details, do citation("pathview") within R.

The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG
license agreement (details at http://www.kegg.jp/kegg/legal.html).
##############################################################################
```

```r
library(gage)
```

```r
library(gageData)

data(kegg.sets.hs)
data(sigmet.idx.hs)
```

```r
kegg.sets.hs = kegg.sets.hs[sigmet.idx.hs]
head(kegg.sets.hs)
```

```
$`hsa00232 Caffeine metabolism`
[1] "10"   "1544" "1548" "1549" "1553" "7498" "9"

$`hsa00983 Drug metabolism - other enzymes`
 [1] "10"     "1066"   "10720"  "10941"  "151531" "1548"   "1549"   "1551"
 [9] "1553"   "1576"   "1577"   "1806"   "1807"   "1890"   "221223" "2990"
[17] "3251"   "3614"   "3615"   "3704"   "51733"  "54490"  "54575"  "54576"
[25] "54577"  "54578"  "54579"  "54600"  "54657"  "54658"  "54659"  "54963"
[33] "574537" "64816"  "7083"   "7084"   "7172"   "7363"   "7364"   "7365"
[41] "7366"   "7367"   "7371"   "7372"   "7378"   "7498"   "79799"  "83549"
[49] "8824"   "8833"   "9"      "978"
```

```
$`hsa00230 Purine metabolism`
  [1] "100"    "10201"  "10606"  "10621"  "10622"  "10623"  "107"    "10714"
  [9] "108"    "10846"  "109"    "111"    "11128"  "11164"  "112"    "113"
 [17] "114"    "115"    "122481" "122622" "124583" "132"    "158"    "159"
 [25] "1633"   "171568" "1716"   "196883" "203"    "204"    "205"    "221823"
 [33] "2272"   "22978"  "23649"  "246721" "25885"  "2618"   "26289"  "270"
 [41] "271"    "27115"  "272"    "2766"   "2977"   "2982"   "2983"   "2984"
 [49] "2986"   "2987"   "29922"  "3000"   "30833"  "30834"  "318"    "3251"
 [57] "353"    "3614"   "3615"   "3704"   "377841" "471"    "4830"   "4831"
 [65] "4832"   "4833"   "4860"   "4881"   "4882"   "4907"   "50484"  "50940"
 [73] "51082"  "51251"  "51292"  "5136"   "5137"   "5138"   "5139"   "5140"
 [81] "5141"   "5142"   "5143"   "5144"   "5145"   "5146"   "5147"   "5148"
 [89] "5149"   "5150"   "5151"   "5152"   "5153"   "5158"   "5167"   "5169"
 [97] "51728"  "5198"   "5236"   "5313"   "5315"   "53343"  "54107"  "5422"
[105] "5424"   "5425"   "5426"   "5427"   "5430"   "5431"   "5432"   "5433"
[113] "5434"   "5435"   "5436"   "5437"   "5438"   "5439"   "5440"   "5441"
[121] "5471"   "548644" "55276"  "5557"   "5558"   "55703"  "55811"  "55821"
[129] "5631"   "5634"   "56655"  "56953"  "56985"  "57804"  "58497"  "6240"
[137] "6241"   "64425"  "646625" "654364" "661"    "7498"   "8382"   "84172"
[145] "84265"  "84284"  "84618"  "8622"   "8654"   "87178"  "8833"   "9060"
[153] "9061"   "93034"  "953"    "9533"   "954"    "955"    "956"    "957"
[161] "9583"   "9615"

$`hsa04514 Cell adhesion molecules (CAMs)`
  [1] "1000"      "1001"      "100133583" "1002"      "1003"      "100506658"
  [7] "1013"      "10666"     "10686"     "1272"      "1364"      "1365"
 [13] "1366"      "137075"    "1462"      "1493"      "149461"    "214"
 [19] "22871"     "23114"     "23308"     "23562"     "23705"     "24146"
 [25] "257194"    "25945"     "26047"     "26285"     "2734"      "29126"
 [31] "29851"     "3105"      "3106"      "3107"      "3108"      "3109"
 [37] "3111"      "3112"      "3113"      "3115"      "3117"      "3118"
 [43] "3119"      "3122"      "3123"      "3125"      "3126"      "3127"
 [49] "3133"      "3134"      "3135"      "3383"      "3384"      "3385"
 [55] "3655"      "3676"      "3680"      "3683"      "3684"      "3685"
 [61] "3688"      "3689"      "3695"      "3696"      "3897"      "4099"
 [67] "4267"      "4359"      "4684"      "4685"      "4756"      "4897"
 [73] "4950"      "49861"     "5010"      "50848"     "51208"     "5133"
 [79] "5175"      "53842"     "54413"     "57502"     "57555"     "57863"
 [85] "5788"      "5792"      "5797"      "5817"      "5818"      "5819"
 [91] "58494"     "6382"      "6383"      "6385"      "6401"      "6402"
 [97] "6403"      "6404"      "652614"    "6614"      "6693"      "6900"
[103] "7122"      "7412"      "80380"     "80381"     "8174"      "83700"
[109] "8506"      "8516"      "9019"      "9071"      "9073"      "9074"
```

```
[115] "9075"      "9076"      "9080"      "90952"     "914"       "920"
[121] "923"       "925"       "926"       "933"       "9369"      "9378"
[127] "9379"      "940"       "941"       "942"       "947"       "958"
[133] "959"       "965"       "9672"      "999"

$`hsa04010 MAPK signaling pathway`
  [1] "10000"     "100137049" "10125"     "10235"     "10368"     "10369"
  [7] "10454"     "10746"     "10912"     "11072"     "11184"     "11221"
 [13] "11261"     "1147"      "115727"    "123745"    "1326"      "1386"
 [19] "1398"      "1399"      "1432"      "1616"      "1647"      "1649"
 [25] "1843"      "1844"      "1845"      "1846"      "1847"      "1848"
 [31] "1849"      "1850"      "1852"      "1950"      "1956"      "2002"
 [37] "2005"      "207"       "208"       "2122"      "2246"      "2247"
 [43] "2248"      "2249"      "2250"      "2251"      "2252"      "2253"
 [49] "2254"      "2255"      "2256"      "2257"      "2258"      "2259"
 [55] "2260"      "2261"      "2263"      "2264"      "22800"     "22808"
 [61] "23118"     "2316"      "23162"     "2317"      "2318"      "2353"
 [67] "23542"     "25780"     "26279"     "26281"     "26291"     "27006"
 [73] "27091"     "27092"     "27330"     "2768"      "2872"      "2885"
 [79] "30814"     "3164"      "3265"      "3303"      "3304"      "3305"
 [85] "3306"      "3310"      "3312"      "3315"      "355"       "3551"
 [91] "3552"      "3553"      "3554"      "356"       "3725"      "3727"
 [97] "3845"      "391013"    "3925"      "408"       "409"       "4137"
[103] "4149"      "4208"      "4214"      "4215"      "4216"      "4217"
[109] "4296"      "4342"      "4609"      "4616"      "468"       "4763"
[115] "4773"      "4776"      "4790"      "4791"      "4803"      "4893"
[121] "4908"      "4909"      "4914"      "4915"      "50487"     "5058"
[127] "5062"      "51295"     "51347"     "5154"      "5155"      "5156"
[133] "5159"      "51701"     "51776"     "5319"      "5320"      "5321"
[139] "5322"      "5494"      "5495"      "5530"      "5532"      "5533"
[145] "5534"      "5535"      "5536"      "5566"      "5567"      "5568"
[151] "5578"      "5579"      "55799"     "5582"      "5594"      "5595"
[157] "55970"     "5598"      "5599"      "5600"      "5601"      "5602"
[163] "5603"      "5604"      "5605"      "5606"      "5607"      "5608"
[169] "5609"      "5613"      "56940"     "57551"     "5778"      "5801"
[175] "5871"      "5879"      "5880"      "5881"      "5894"      "5906"
[181] "5908"      "5921"      "5922"      "5923"      "5924"      "59283"
[187] "59284"     "59285"     "5970"      "5971"      "6195"      "6196"
[193] "6197"      "6237"      "627"       "6300"      "63928"     "6416"
[199] "64600"     "6654"      "6655"      "6722"      "673"       "6788"
[205] "6789"      "6885"      "7040"      "7042"      "7043"      "7046"
[211] "7048"      "7124"      "7132"      "7157"      "7186"      "7189"
[217] "773"       "774"       "775"       "776"       "777"       "778"
```

```
[223] "7786"      "779"       "781"       "782"       "783"       "784"
[229] "785"       "7850"      "786"       "7867"      "8074"      "80824"
[235] "81579"     "836"       "8398"      "8399"      "84647"     "84867"
[241] "8491"      "8517"      "8550"      "8569"      "8649"      "8681"
[247] "8817"      "8822"      "8823"      "8911"      "8912"      "8913"
[253] "8986"      "9020"      "9064"      "9175"      "9252"      "9254"
[259] "9261"      "929"       "9344"      "93589"     "9448"      "9479"
[265] "9693"      "994"       "9965"      "998"

$`hsa04012 ErbB signaling pathway`
 [1] "10000"  "1026"   "1027"   "10298"  "10718"  "1398"   "1399"   "145957"
 [9] "1839"   "1950"   "1956"   "1978"   "2002"   "2064"   "2065"   "2066"
[17] "2069"   "207"    "208"    "23533"  "23624"  "2475"   "25"     "2549"
[25] "25759"  "27"     "2885"   "2932"   "3084"   "3265"   "369"    "3725"
[33] "374"    "3845"   "399694" "4609"   "4690"   "4893"   "5058"   "5062"
[41] "5063"   "5290"   "5291"   "5293"   "5294"   "5295"   "5296"   "5335"
[49] "53358"  "5336"   "5578"   "5579"   "5582"   "5594"   "5595"   "5599"
[57] "5601"   "5602"   "5604"   "5605"   "5609"   "56924"  "57144"  "572"
[65] "5747"   "5894"   "6198"   "6199"   "6416"   "6464"   "6654"   "6655"
[73] "6714"   "673"    "6776"   "6777"   "685"    "7039"   "815"    "816"
[81] "817"    "818"    "8440"   "8503"   "867"    "868"    "9542"
```

```
foldchanges = res$log2FoldChange
names(foldchanges) = res$entrez
head(foldchanges)
```

```
       7105       64102        8813       57147       55732        2268
-0.35070302          NA  0.20610777  0.02452695 -0.14714205 -1.73228897
```

```
keggres = gage(foldchanges, gsets=kegg.sets.hs)
attributes(keggres)
```

```
$names
[1] "greater" "less"    "stats"
```

```
head(keggres$less,5)
```

```
                                                  p.geomean stat.mean
hsa04672 Intestinal immune network for IgA production 0.006043452 -2.560547
hsa04340 Hedgehog signaling pathway                   0.013323955 -2.248547
hsa04916 Melanogenesis                                0.030996739 -1.877209
hsa04972 Pancreatic secretion                         0.037135109 -1.794718
hsa04612 Antigen processing and presentation          0.044082096 -1.716738
                                                          p.val     q.val
hsa04672 Intestinal immune network for IgA production 0.006043452 0.9763115
hsa04340 Hedgehog signaling pathway                   0.013323955 0.9763115
hsa04916 Melanogenesis                                0.030996739 0.9763115
hsa04972 Pancreatic secretion                         0.037135109 0.9763115
hsa04612 Antigen processing and presentation          0.044082096 0.9763115
                                                      set.size        exp1
hsa04672 Intestinal immune network for IgA production       47 0.006043452
hsa04340 Hedgehog signaling pathway                         56 0.013323955
hsa04916 Melanogenesis                                     101 0.030996739
hsa04972 Pancreatic secretion                              101 0.037135109
hsa04612 Antigen processing and presentation                75 0.044082096
```

```
pathview(gene.data=foldchanges, pathway.id="hsa05310")
```

```
'select()' returned 1:1 mapping between keys and columns
```

```
Info: Working in directory /Users/chrisz/Desktop/BGGN 213/week 6/class12
```

```
Info: Writing image file hsa05310.pathview.png
```

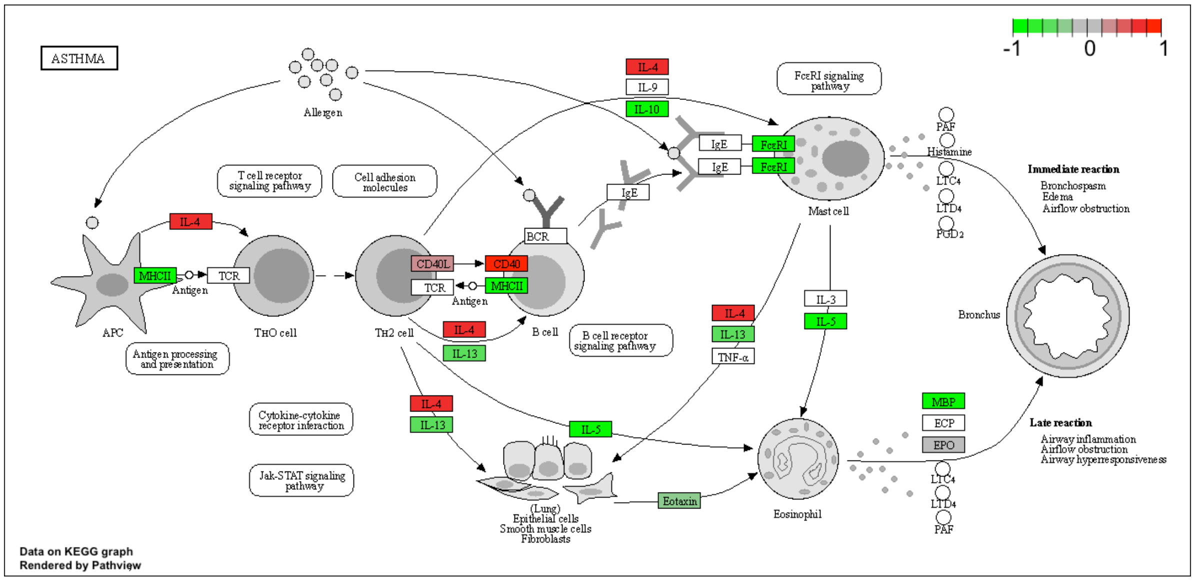


Figure 1: The asthama pathway with hsa05310 gene